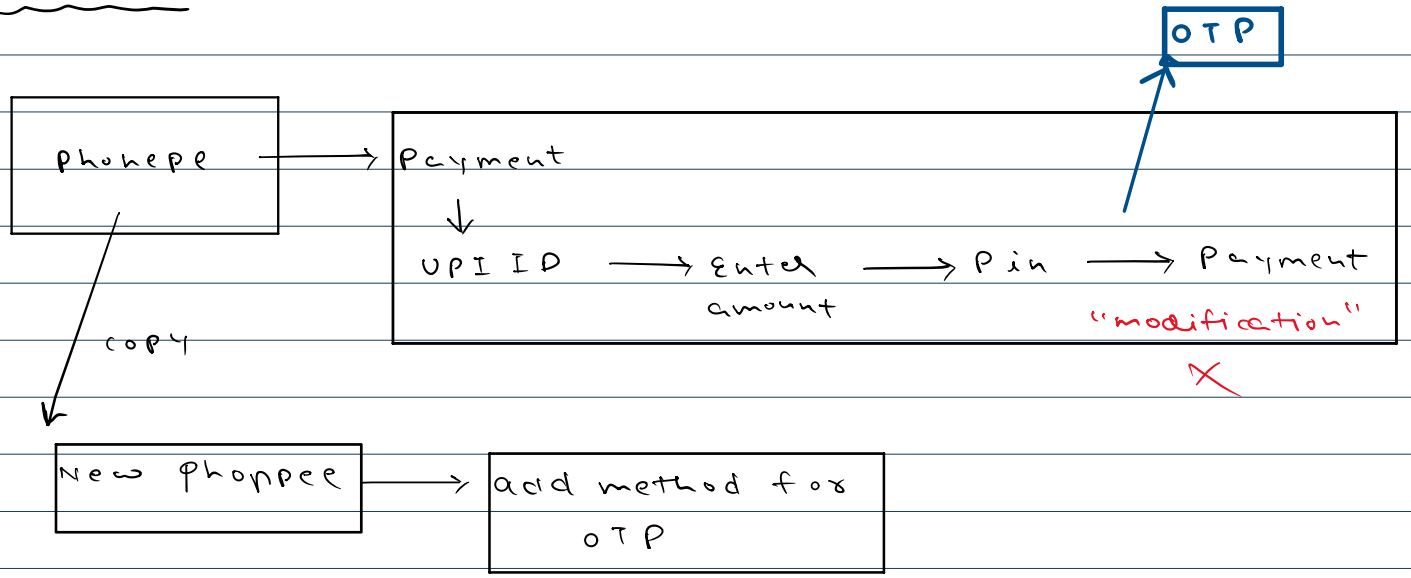
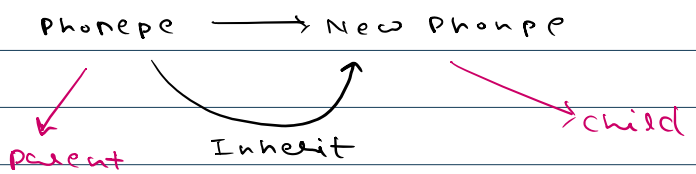
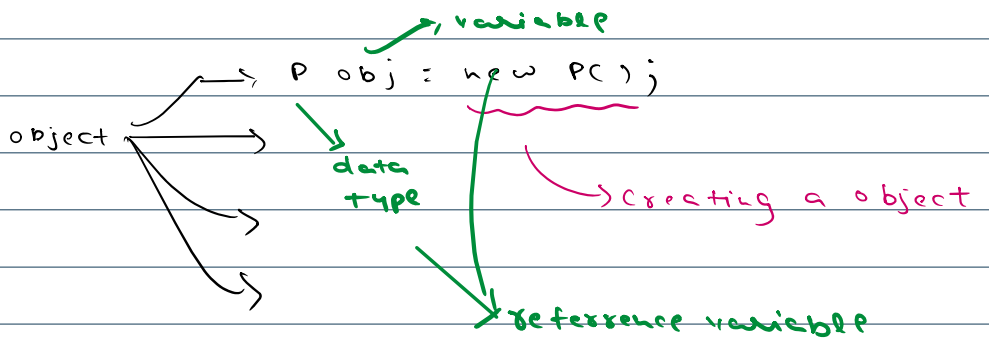
Inheritance





class child. extends parent

↳ all the properties of parent will be there in child



P obj = new C();

↳ parent class reference

↳ object of child class

variables → Reference type (Parent class)

{ Compile Time }

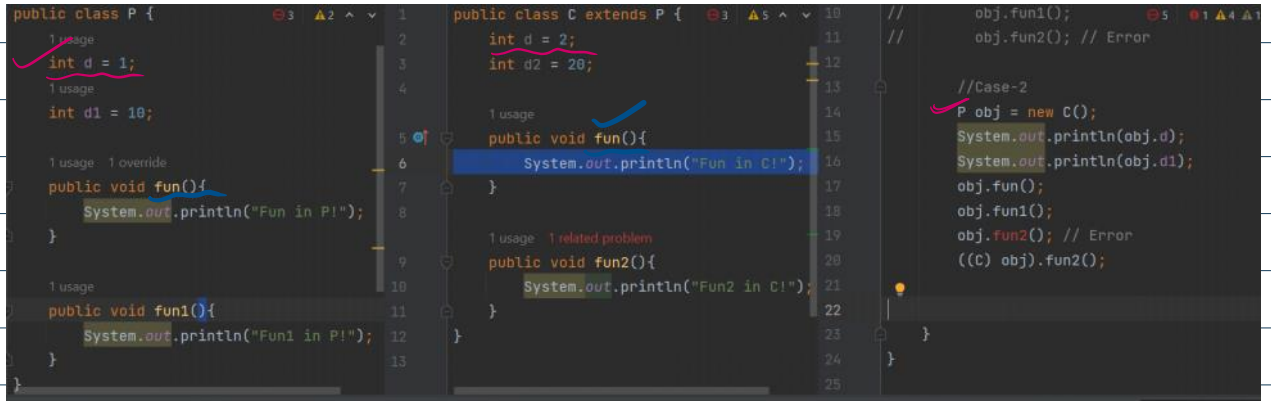
method → Based on the

parent class reference variable

object of child class

Method resolution → Based on the object type

{ Runtime }



```
public class P {
    int d = 1;
    int d1 = 10;

    public void fun(){
        System.out.println("Fun in P!");
    }

    public void fun1(){
        System.out.println("Fun1 in P!");
    }
}
```

```
public class C extends P {
    int d = 2;
    int d2 = 20;

    public void fun(){
        System.out.println("Fun in C!");
    }

    public void fun2(){
        System.out.println("Fun2 in C!");
    }
}
```

```
//      obj.fun1();
//      obj.fun2(); // Error

    //Case-2
    P obj = new C();
    System.out.println(obj.d);
    System.out.println(obj.d1);
    obj.fun();
    obj.fun1();
    obj.fun2(); // Error
    ((C) obj).fun2();
}
```

d	d1
~~d~~	~~d2~~

variables → Parent

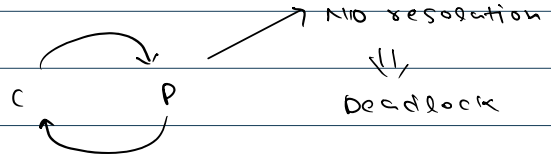
methods → Child

C obj = new P();

reference of child → object of parent

C obj = new C();

reference of child → object of child



Types of Inheritance

1- Single Inheritance

class Parent { class Child extends Parent {

2)- Multi level Inheritance

class A { abc, pqr }

class B extends A { xyz, tuv } → abc, pqr

class C extends B { mno } → abc, pqr, xyz, tuv

3)- Hierarchical Inheritance

class Parent { abc, pqr }

class Child1 extends Parent { mno } → abc, pqr

class Child2 extends Parent { tuv } → abc, pqr

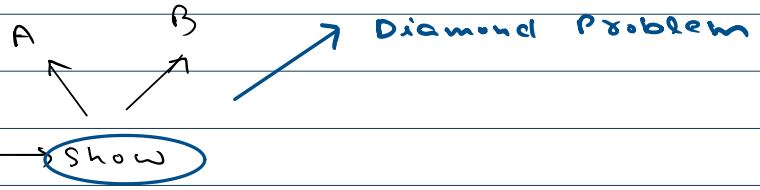
4)- Multiple Inheritance

class A { show() }

class B { show() }

class C extends A, B { }

Jvm doesn't allow



Compile Time Polymorphism

method overloading